PCT/EP 00 / 02440

Europäisches Patentamt | European Patent Office | Office européen des brevets

EP00 / 02440

4

REC'D 19 MAY 2000

WIPO PCT

# Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

**Patentanmeldung Nr. | Patent application No. | Demande de brevet n°**

99105680.5

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 10/05/00
LA HAYE, LE

Best Available Copy

THIS PAGE BLANK (USPTO)

Europäisches Patentamt | European Patent Office | Office européen des brevets

# Blatt 2 der Bescheinigung
# Sheet 2 of the certificate
# Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99105680.5

Anmeldetag:
Date of filing:
Date de dépôt: 19/03/99

Anmelder:
Applicant(s):
Demandeur(s):
SIEMENS AKTIENGESELLSCHAFT
80333 München
GERMANY

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Data communications apparatus and method of communicating data

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

| Staat: State: Pays: | Tag: Date: Date: | Aktenzeichen: File no. Numéro de dépôt: |
|---|---|---|

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

EPO - Munich
17
19. März 1999



Description of invention:

Data communications apparatus and method of communicating data

The present invention relates to data communications apparatus and methods for communicating data. More specifically, the present invention relates to data communications apparatus and methods of communicating data in which data is punctured or. The transmitted elements may be bits.

Digital communications systems are arranged to communicate data by representing the data in a form which facilitates transmission of the data via a medium through which communication is effected. For example, in a case of radio communications, the data is represented as radio signals and transmitted between transmitters and receivers of the communications system via the ether. In the case of broadband telecommunications networks, the data may be represented as light and communicated via, for example, a fibre optic network between transmitters and receivers of the system.

During transmission of data, bits or symbols of the communicated data can be corrupted to the effect that these bits or symbols can not be correctly determined at the receiver. For this reason, the data communications systems often include means for mitigating the corruption of the data which occurs during transmission. One of these means is to provide transmitters of the system with encoders, which encode the data prior to transmission, in accordance with an error control code. The error control code is arranged to add redundancy to the data in a controlled way. At the receiver, errors occurring during transmission may be corrected by decoding the error control code, thereby recovering the original data. The decoding is effected using an error decoding algorithm corresponding to the error control code, which is known to the receiver.

After data has been encoded, there is often a requirement to puncture data bits or symbols from a block of encoded data before transmission of that data. The term puncturing as used herein refers to a process of cancelling or deleting bits from an encoded data block to the effect that the punctured bits are not transmitted with that data block. Puncturing might be required because, for example, a multiple access scheme, which serves to effect communication of the data via the data bearing media, requires the data to formatted into blocks having a pre-determined size, which does not correspond to the size of the encoded data frame. In order to fit the encoded data frame into the transport data block of the pre-determined size, therefore, data bits from the encoded data frame are either punctured, to decrease the size of the encoded data block, in a case where the encoded data frame is larger than the size of the transport block, or repeat bits of the encoded data frame in a case where the encoded data frame is smaller than the pre-determined size of the transport block.

As will be appreciated, the data frames may be transmitted un-encoded in the transport data block. In this case, it is not appropriate to puncture the data frame in order to fit the data frame into the transport data block, a plurality of transport data blocks must be used to convey the data frame. In a case where the data frame is smaller than the transport data block, then the data bits or symbols are repeated to an extent necessary to fill the remainder of the transport data block.

As is familiar to those skilled in the art, an effect of puncturing an encoded data frame, is that a probability of correctly recovering the original data is reduced. Furthermore, the performance of known error control codes and decoders for these error control codes is best when the errors occurring during transmission of the data are caused by Gaus-

sian noise, with an effect that the errors are independently distributed throughout the transport data block. Similarly, therefore, if an encoded data frame is to be punctured, the positions within the encoded data frame at which bits are punctured, should be separated from each other as far as possible. As such, the puncturing positions should be evenly distributed throughout the frame. Similarly, because errors during transmission often occur in bursts, particularly in the case of radio communications systems which do not employ interleaving, positions within an encoded or an un-encoded data frame, at which data bits are to be repeated, should be arranged to be evenly separated throughout the frame.

Known methods of selecting the positions of bits or symbols to be punctured or repeated within an encoded data frame, include dividing the number of bits or symbols within a frame, by the number of bits or symbols to be punctured and selecting positions at integer values corresponding to the division. However, in a case where the number of bits to be punctured is not an integer division of the frame, an equidistant separation of punctured or repeated positions does not result, providing the disadvantage that some positions may be closer than this integer, and in some cases even adjacent one another.

The interleaving in the transport mulitplexing scheme is performed in two steps. The different solutions have some implications in uplink. It is believed that puncturing will be useful also in the uplink, for example in order to avoid multicode. There is then a potential problem since if FS-MIL is used in the uplink multiplexing scheme (Figure 1) together with the current rate matching algorithm, the performance could be degraded.

Consider, as an example, a case where layer 2 delivers a transport block with 160 bits on a transport channel with

transmission time interval 80 ms. This means that after the 1st interleaver, the data will be interleaved over 8 frames as illustrated in figure 2. Now assume that four bits in each frame should be punctured in order to balance the quality of service requirement of this transport channel with other channels. The result of the rate matching algorithm (with $e=2N_c$) is that bit 4, 9, 14, and 19 (index starting at 0) in each frame should be punctured. In figure 2, a punctured bit is illustrated with bold font. Consequently, 8 adjacent bits will be punctured which is clearly undesirable.

One obvious way of avoiding this would be to shift the puncturing pattern in each frame. Denote the number of bits in one frame before rate matching by $N_i$, the number of bits after rate matching by $N_c$, the index to the punctured/repeated bit by $m_j$, the frame number by k, and the number of interleaved frames by K. Consider the case when $N_i>N_c$, i.e. puncturing. In the example above $N_i=20$, $N_c=16$, $m_1=4$, $m_2=9$, $m_3=14$, $m_4=19$, $k=1\ldots7$, and $K=8$. Shifting could then be achieved with the following formula:

$m_{j\ shifted} = (m_j + k*\lceil N_c/(N_i-N_c)/K\rceil) \bmod N_i$, *where* $\lceil\ \rceil$ *means round upwards.*

The same example as before would then yield the result in figure 3.

As seen in the figure, puncturing adjacent bits is avoided to some extent. However, there is a wrap around effect, i.e. bit 43 and 44 are for example both punctured. If the puncturing ratio is low the probability of puncturing adjacent bits decreases. In figure 4, an example with 10% puncturing is illustrated. As seen in the figure adjacent bits are still punctured. Hence, it is possible that a loss in performance will result.

If the 1st interleaver is optimized and the 2nd interleaver is kept simple, puncturing does no longer need the rate-matching algorithm described in. An optimized 1st interleaver should

reorder the bits so that adjacent bits are separated. Consequently, puncturing can easily be performed by removing consecutive bits after interleaving. However, there are two possibilities. Consider the scenario illustrated in figure 5.

The 4 blocks on TrCH A are interleaved together and rate matching is then applied. If puncturing is used, consecutive bits in each frame are removed. It is therefore very unlikely that punctured bits in one frame were adjacent after coding. However, there is no guarantee that punctured bits in *different frames* were not adjacent after coding. Consequently, there might be a performance loss if this approach is used.

An alternative is to only puncture consecutive bits once every transmission time interval. The drawback of this approach is that at time 30 ms, bits on TrCH A will be repeated since no data is present on TrCH B. It would probably have been better to decrease the amount of puncturing rather than repeat some other bits. This issue has already been discussed and was one of the motives for combining the static and dynamic rate matching. However, there are still benefits with the combined rate matching even if this approach would be chosen. The transmission of non real time (NRT) transport blocks is still possible if some modifications are made to the original NRT concept. In the original proposal it was possible to increase the puncturing and thereby make room for NRT block but this would not be possible in this new approach. In the above example the restriction would be that the length of the NRT block(s) would have to be shorter or of equal length to the transport blocks of TrCH B. However, in cases where repetition is used, the number of repeated bits can of course be decreased in order to make room for NRT blocks.

It was pointed out the problem for puncturing if FS-MIL is used in uplink multiplexing scheme. This problem was occurred due to the results of re-ordering between rate matching and 1st interleaving

If current rate matching algorithm is applied for output of inter-frame FS-MIL, the plural adjacent bits in specific row will be punctured as shown in Figure 2. In order to avoid this, the shifting of the puncturing patterns is then introduced in Figure 3. However, some puncturing adjacent bits are still remained due to a wrap around effect and it is concerned these puncturing will cause some performance degradations.

To solve the above problem, the following modification for current rate matching could be effective: i.e. puncturing with simple shifting rule before column randomizing of inter-frame FS-MIL (To easily understand the main characteristic of the processing block, the name of "row-by-row processing" is renamed "column randomizing".)

Figure 6 shows an example of puncturing patterns when this modification is applied for the same bit sequence example as before. Rate matching with shifting is done just after 1st stage block interleaving. In this figure, the puncturing adjacent bits could not be seen any more. Therefore, the performance loss should not be occurred due to such puncturing.

It is not necessary to perform the above rate matching before column randomizing, actually. The equivalent rate matching could be done after column randomizing with taking the column randomizing rule into account and this could be easily achieved only replacing the initial offset value of puncturing with a simple formula. The exact modified rate-matching algorithm is shown in List 1. In this list, $e_{offset}$ is introduced to set initial offset of each frame for uplink rate matching. Furthermore, $e_{offset}$ is not only applied for puncturing but also repetition. This could also place repetition bits more uniformly.

The interleaving in the transport multiplexing scheme is performed in two steps. As explained in the sections above implications of the different solutions have some implications in uplink.

In the following it is shown that the proposed solutions, i.e.puncturing pattern is still not optimum in all cases and a further modification is suggested to arrive at an algorithm which works satisfactory in all cases.

One embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings wherein;

FIGURE 7 is a schematic block diagram of a mobile radio communication system;

FIGURE 8 is a schematic block diagram of a data communications apparatus forming a link between the mobile station and a base station of the communications network shown in Figure 1;

FIGURE 9 *1st interleaving of 80 ms and 1:8 puncturing with improved algorithm*

FIGURE 10 *Principle of optimised puncturing*

FIGURE 11 Lookup table

FIGURE 12 *1st interleaving of 80 ms and 1:5 puncturing*

FIGURE 13 *1:8 puncturing with proposed algorithm*

FIGURE 14 *Unequal number of bits per frame*

FIGURE 15 Puncturing pattern

An example embodiment of the present invention will be described with reference to a mobile radio communications system. Mobile radio communications systems are provided with multiple access systems which operate, for example, in accordance with Time Division Multiple Access (TDMA) such as that used in the Global System for Mobiles, which is a mobile radio telephone standard administered by the European Telecommunications Standards Institute. The mobile radio communications system, alternatively, could be provided with a multiple access system which operates in accordance with Code Division Multiple Access (CDMA) such as that proposed for the third generation Universal Mobile Telecommunication System. However, as will be appreciated, any data communications system could be used to illustrate an example embodiment of the present invention, such as a Local Area Network, or a Broadband Telecommunications Network operating in accordance with Asynchronous Transfer Mode. These illustrative example data communications systems are characterised in particular in that data is transmitted as bursts, packets or blocks. In the case of a mobile radio communications system, the data is transported in bursts of data bearing radio signals, which represent a pre-determined data size. An example of such a mobile radio communication system is shown in Figure 7.

In Figure 7, three base stations BS, are shown to communicate radio signals with mobile stations MS, within a radio coverage area formed by cells 1 defined by broken lines 2. The base stations BS, are coupled together using a network interworking unit NET. The mobile stations MS, and the base stations BS communicate data by transmitting radio signals designated 4, between antennas 6, coupled to the mobile stations MS and the base stations BS. The data is communicated between the mobile stations MS and the base stations BS using a data communications apparatus in which the data is transformed into the radio signals 4, which are communicated to the receive

antenna 6, which detects the radio signals. The data is recovered by the receiver from the radio signals.

An illustrative example of a data communications apparatus forming a radio communications link between one of the mobile stations MS, and one of the base stations BS, is shown in Figure 8, where parts also appearing in Figure 7 bear identical numerical designations. In Figure 8 a source of data 10, generates data frames 8, at a rate determined by a type of data which the source is generating. The data frames 8, generated by the source 10, are fed to a rate converter 12, which operates to convert the data frames 8, into transport data blocks 14. The transport data blocks 14, are arranged to be substantially equal in size, to a pre-determined size of an amount of data which can be carried by bursts of data bearing radio signals via which data is communicated by a radio interface formed by a transmitter 18, and receiver 22, pair.

The data transport block 14 is fed to a radio access processor 16, which operates to schedule transmission of the transport data block 14, over the radio access interface. At an appropriate time the transport data block 14, is fed by the radio access processor 16, to a transmitter 18 which operates to convert the transport data block into the burst of data bearing radio signals which are transmitted in a time period allocated for the transmitter to effect communication of the radio signals. At the receiver 22, an antenna 6'' of the receiver detects the radio signals and down converts and recovers the data frame which is fed to a radio access de-scheduler 24. The radio access de-scheduler 24, feeds the received data transport block to a rate de-converter 26, under control of the multiple access de-scheduler 24, effected via a conductor 28. The rate de-converter 26, thereafter feeds a representation of the regenerated data frame 8, to a destination or sink for the data frame 8 which is represented by the block 30.

The rate converter 12, and rate de-converter 26, are arranged to make, as far as possible, optimum use of the data bearing capacity available within the transport data block 14. This is effected in accordance with the illustrative embodiment of the present invention by the rate matching converter 12, which operates to encoded the data frame, and then puncture or repeat data bits or symbols selected from the encoded data frame, to the effect of generating a transport data block, which fits with the size of the data blocks 14. The rate converter 12, has an encoder, and a puncturer. The data frame 8, fed to the encoder, is encoded to generate an encoded data frame, which is fed to the puncturer. The encoded data frame is then punctured by the puncturer, to generate the data transport block 14.

The assumption has been that puncturing is allowed in both uplink and downlink. When merging the ETSI and ARIB specifications, ARIB's assumption of no puncturing in uplink was put in. It is believed that puncturing will be useful also in the uplink, for example in order to avoid multicode. There is then a potential problem since if FS-MIL is used in the uplink multiplexing scheme together with the current rate matching algorithm, the performance could be degraded. This has been shown in considering, as an example, a case where layer 2 delivers a transport block with 160 bits on a transport channel with transmission time interval 80 ms and assuming that four bits in each frame should be punctured. The result is that 8 adjacent bits will be punctured which is clearly undesirable.

The proposal was to shift the puncturing pattern in each frame. This is equivalent to applying the puncturing before the column shuffling, even if it is actually performed after inter frame interleaving. Indeed, in the above mentioned example, there are no more adjacent bits punctured, as shown in.

However, there exist still cases, where adjacent bits are being punctured, depending on the puncturing rate. Consider e.g. the case, where $N_i=16$, $N_c=14$, $m_1=4$, $m_2=14$, $k=1...7$, and $K=8$. For simplicity, only the field before interleaving is shown in Fig. 9. As can be seen, adjacent bits 31-32 and 95-96 are punctured which is clearly undesirable.

The goal of a good puncturing algorithm is to spread punctured bits evenly as possible. This was the driving principle for the algorithm in as well. This can best be obtained by puncturing every $n^{th}$ bit (for non integer puncturing rates sometimes every $n^{th}$ and sometimes every $n+1^{st}$ bit). We can try to apply this principle also for puncturing after interleaving, but there is one constraint: We have to distribute punctured bits on all frames evenly. For example, assume 80 ms interleaving and a puncturing rate of 1:6. By puncturing every $6^{th}$ bit we would only puncture column 0,2,4,6 but not 1,3,5,7 which is of course impossible. To balance puncturing between columns, we have to change the puncturing interval sometimes (here once) to avoid hitting always the same columns. This is shown in Fig. 10. Bold horizontal arrows show puncturing distance of 6 and the thick hollow arrow shows puncturing distance 5 to avoid hitting the first column twice. After having punctured every column once, the pattern can be shifted down by 6 rows to determine the next bits to be punctured. Obviously this is equivalent to puncturing every $6^{th}$ bit in each column and shifting puncturing patterns in different columns relative to each other as already proposed in [5, 6, 7], but the amount of column shifting is now determined differently.

We now present the formulas for the optimised algorithm: Denote the number of bits in one frame before rate matching by $N_i$, the number of bits after rate matching by $N_c$, the index to the punctured/repeated bit by $m_j$, the frame number by $k$, and the number of interleaved frames by $K$. We mainly consider the case when $N_i>N_c$, i.e. puncturing, but the formulars will

be applicable for repetition as well. In the example above $N_i$=20, $N_c$=16, $m_1$=4, $m_2$=9, $m_3$=14, $m_4$=19, k=1…7, and K=8. Shifting could then be achieved with the following formula:

-- calculate average puncturing distance

q:= $(\lfloor N_c/(|N_i-N_c|)\rfloor)$ mod K -- *where* $\lfloor\ \rfloor$ *means round downwards and* $|\ |$ *means absolute value* .

Q:= $(\lfloor N_c/(|N_i-N_c|)\rfloor)$ div K

if q is even -- *handle special case:*

then q = q - 1 / lcd(q, K) -- where lcd (q, K) means largest common divisor of q and K

-- note lcd can be easily computed using bit manipulations, because K is a power of 2.

-- for the same reason calculations with q can be easily done using binary fixed point

-- arithmetic (or integer arithmetic and a few shift operations).

endif

- calculate S and T, S represents the shift of the row mod K and T the shifting amount div K

for *i = 0* to *K-1*

$S(R_K(\lceil i*q\rceil \bmod K)) = (\lceil i*q\rceil \text{ div } K)$ -- *where* $\lceil\ \rceil$ *means round upwards.*

$T((R_K(\lceil i*q\rceil \bmod K)) = i$ -- $R_K(k)$ *reverts the interleaver as in [7]*

end for

In a real implementation, these formulas can be implemented as a lookup table as shown in figure 11. The table also includes the effect of re mapping the column randomising achieved by $R_K(k)$. Obviously S can also be calculated from T as jet an other implementation option.

Then, $e_{offset}$ can be calculated as

$$e_{offset}(k) = ((2*S) + 2*T*Q + 1)*y + 1) \bmod 2Nc$$

$e_{offset}(k)$ is then used to pre load $e$ in the rate matching formula in [2].

This algorithm will obtain the perfect puncturing as if puncturing using the rate matching algorithm was applied directly

before interleaving, if the puncturing rate is an odd fraction i.e. 1:5 or 1:9. For other cases, adjacent bits will never be punctured, but one distance between punctured bits may be larger by up to lcd(q,K)+1 than the other ones. Note that this algorithm should be applied to bit repetition as well as already suggested in [7]. While repeating adjacent bits is not as bad as puncturing them, it is still advantageous to distribute repeated bits as evenly as possible.
The basic intention of these formulas is to try to achieve equidistant spacing of the punctured bits in the original order, but taking into account the constraint, that the bits have to be punctured equally in different frames. This may make it necessary to reduce the puncturing distance by 1 sometimes. The presented algorithm is optimum in the sense, that it will never reduce the distance by more than 1, and will reduce it only as often as necessary. This gives the best possible puncturing pattern under the above mentioned constraints.
The following is an example using the first set of parameters i.e. puncturing by 1:5 (Fig. 12). Obviously the optimised algorithm not only completely avoids puncturing adjacent bits, it also distributes punctured bits with equal spacing in the original sequence. In fact the same properties are achieved, as if the puncturing had been done directly after coding before interleaving.

Let us now investigate the next case i.e. puncturing by 1:8 (Fig. 13). Again puncturing of adjacent bits is avoided. In this case it is not possible to obtain an equidistant puncturing because then all bits of one single frame would be punctured, which is totally unacceptable. In this case most of the distances between adjacent bits are 7 (only one less than would be the case with an optimum distribution). Some distances are larger (every eighth) in exchange.

There are two cases, where the rate matching can change during the transmission time interval:

a) The number of input bits Ni is not divisible by K. Then the last frames will carry one bit less than the first ones and therefore also have a slightly lower puncturing rate. Note that it is not clear, whether this case will be allowed or whether the coding will be expected to deliver a suitable number.

b) Due to fluctuations in other services which are multiplexed on the same connection the puncturing can be relaxed in later frames.

In these cases the balanced puncturing scheme could still suffer. Due to the unpredictable nature of case b) it seems unlikely, that any scheme can be found, which could lead to a near perfect puncturing pattern, so here we may have to live with some unpredictable behaviour anyhow. In case a) however, we propose not to change the puncturing pattern in the last rows. Instead we suggest to use the same puncturing algorithm as for the first columns, but simply omit the last puncture. Consider as an example that 125 input bits are to be punctured to give 104 output bits, interleaved over 8 frames. Then the puncturing pattern would look like shown in Fig. 14. The last columns have one less input bit than the first ones, by omitting the last puncture, the columns all have 13 bits.

There also is an alternative proposal to use an optimised $1^{st}$ interleaver, and use a simple $2^{nd}$ interleaver and a simple puncturing scheme. This relies on the expectation, that an optimised interleaver will distribute bits in a way that puncturing blocks of bits after interleaving will spread these punctured bits evenly before interleaving. However, the experience with puncturing after a simple $1^{st}$ interleaver tells, that this is not an easy task. As the single interleaver can not be optimised for all puncturing rates it is next to impossible, that good properties can be achieved: The reason is as follows: The puncturing patterns (Fig 15) for n+1 bits must be identical to the puncturing pattern for n bits, but one additional bit can be selected for puncturing. If the puncturing pattern for n bits is good (see firs row in

the table below), then what ever bit is punctured to get n+1 bits (second row), it is impossible to reach an optimum distribution of n+1 bits (last row).

Further more such an interleaver would have to be a compromise between good puncturing properties for block puncturing and good general interleaving properties at the same time. Finding a scheme that optimally satisfies both constraints seems impossible.

Concluding we think that such a optimised 1st interleaver will unfortunately not exist, so we have to use the other alternative i.e. puncturing after a simple 1st interleaver followed by a second interleaver with optimised interleaving properties.

Thus near optimum puncturing patterns are possible when applying rate matching after first interleaving. The necessary algorithm is not very complex, it is similar to the puncturing algorithm itself but has to be executed once per frame only, not once per bit.

THIS PAGE BLANK (USPTO)

EPO - Munich
17
19. März 1999



Claims:

1. Method of communicating data frames, whereby the transmitted elements are distributed on one or several frames by using an interleaver and where elements are punctured or repeated wherein the puncturing or repetition is done in a way that the pattern, when related to the original ordering of the elements before interleaving avoids puncturing/repeating adjacent elements or elements not far away from each other.

2. Method of communicating data frames, whereby the transmitted elements are distributed on one or several frames by using an interleaver and where elements are punctured or repeated wherein the puncturing or repetition iß done in a way that the pattern, when related to the original ordering of the elements before interleaving will be equidistant or roughly equidistant.

3. Method of communicating data frames, whereby the transmitted elements are distributed on one or several frames by using an interleaver and where elements are punctured or repeated wherein the puncturing or repetition pattern occurring within the frames is shifted relative to the first frame in a way that the resulting puncturing or repetition pattern, when related to the original ordering of the elements before interleaving will be equidistant or roughly equidistant.

4. Method of communicating data frames, as claimed in any of the preceding claims, whereby the puncturing/repetition rate is an integer fraction (1/p) whereby p and the number of frames k do not have a common divisor, whereby the patterns occurring within the frames is shifted relative to the first frame in a way that the resulting puncturing or repetition pattern, when related to the original ordering of the elements before interleaving is equidistant.

5. Method of communicating data frames, as claimed in any of the preceding claims, whereby the puncturing/repetition rate is NOT an integer fraction (1/p) or p and the number of frames k do not have a common divisor, whereby the patterns occurring within the frames is shifted relative to the first frame by applying the relative shifts that would be used for the next higher puncturing rate which fulfils the precondition of the preceding claim.

6. Method of communicating data frames, as claimed in any of the preceding claims, whereby the number of elements for puncturing/repetition is NOT identical in all frames, whereby the same patterns are used as in the preceding claims but some of the puncturing/repetition is not performed.

7. Method of communicating data frames, as claimed in any of the preceding claims, whereby the number of elements for puncturing/repetition is NOT identical in all frames, whereby the same patterns are used as in the preceding claims but puncturing/repetition is not performed for the first or last elements.

8. Method of communicating data frames, as claimed in any of the preceding claims, whereby puncturing is performed

9. Method of communicating data frames, as claimed in any of the preceding claims, whereby repetition is performed

10. Method of communicating data frames, as claimed in any of the preceding claims, whereby the elements are binary digits

11. Method of communicating data frames, as claimed in any of the preceding claims, whereby the frames have a duration of 10 ms and interleaving is done over a power of two of frames

12. Method of communicating data frames, as claimed in any of the preceding claims, whereby the frames are transmitted using a CDMA transmission system

13. Data communications apparatus which operates to communicate data frames, said apparatus comprising means for communicating data frames as claimed in in any of the preceding claims.
claims

THIS PAGE BLANK (USPTO)

EPO - Munich
17
19. März 1999

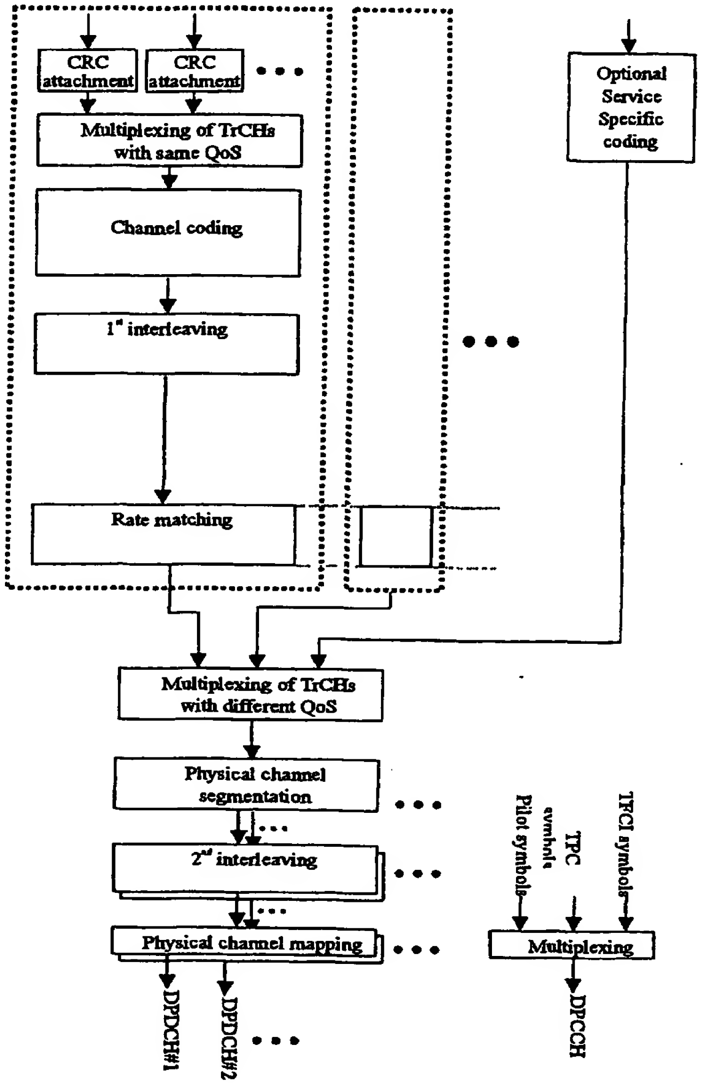


FIG 1

Bit sequence

| |
|---|
| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ... 159 |

Row by row processing
8[4[2x2]x2]

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 |
| 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 |
| 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 |
| 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |

| 0 | 4 | 2 | 6 | 1 | 5 | 3 | 7 |
|---|---|---|---|---|---|---|---|
| 8 | 12 | 10 | 14 | 9 | 13 | 11 | 15 |
| 16 | 20 | 18 | 22 | 17 | 21 | 19 | 23 |
| 24 | 28 | 26 | 30 | 25 | 29 | 27 | 31 |
| 32 | 36 | 34 | 38 | 33 | 37 | 35 | 39 |
| 40 | 44 | 42 | 46 | 41 | 45 | 43 | 47 |
| 48 | 52 | 50 | 54 | 49 | 53 | 51 | 55 |
| 56 | 60 | 58 | 62 | 57 | 61 | 59 | 63 |
| 64 | 68 | 66 | 70 | 65 | 69 | 67 | 71 |
| 72 | 76 | 74 | 78 | 73 | 77 | 75 | 79 |
| 80 | 84 | 82 | 86 | 81 | 85 | 83 | 87 |
| 88 | 92 | 90 | 94 | 89 | 93 | 91 | 95 |
| 96 | 100 | 98 | 102 | 97 | 101 | 99 | 103 |
| 104 | 108 | 106 | 110 | 105 | 109 | 107 | 111 |
| 112 | 116 | 114 | 118 | 113 | 117 | 115 | 119 |
| 120 | 124 | 122 | 126 | 121 | 125 | 123 | 127 |
| 128 | 132 | 130 | 134 | 129 | 133 | 131 | 135 |
| 136 | 140 | 138 | 142 | 137 | 141 | 139 | 143 |
| 144 | 148 | 146 | 150 | 145 | 149 | 147 | 151 |
| 152 | 156 | 154 | 158 | 153 | 157 | 155 | 159 |

Radio frame #1 Radio frame #2 ... Radio frame #8

FIG2

Bit sequence

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ... 159

Row by row processing
8[4[2x2]x2]

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 |
| 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 |
| 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 144 | 145 | 146 | 147 | 48 | 149 | 150 | 151 |
| 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 4 | 2 | 6 | 1 | 5 | 3 | 7 |
| 8 | 12 | 10 | 14 | 9 | 13 | 11 | 15 |
| 16 | 20 | 18 | 22 | 17 | 21 | 19 | 23 |
| 24 | 28 | 26 | 30 | 25 | 29 | 27 | 31 |
| 32 | 36 | 34 | 38 | 33 | 37 | 35 | 39 |
| 40 | 44 | 42 | 46 | 41 | 45 | 43 | 47 |
| 48 | 52 | 50 | 54 | 49 | 53 | 51 | 55 |
| 56 | 60 | 58 | 62 | 57 | 61 | 59 | 63 |
| 64 | 68 | 66 | 70 | 65 | 69 | 67 | 71 |
| 72 | 76 | 74 | 78 | 73 | 77 | 75 | 79 |
| 80 | 84 | 82 | 86 | 81 | 85 | 83 | 87 |
| 88 | 92 | 90 | 94 | 89 | 93 | 91 | 95 |
| 96 | 100 | 98 | 102 | 97 | 101 | 99 | 103 |
| 104 | 108 | 106 | 110 | 105 | 109 | 107 | 111 |
| 112 | 116 | 114 | 118 | 113 | 117 | 115 | 119 |
| 120 | 124 | 122 | 126 | 121 | 125 | 123 | 127 |
| 128 | 132 | 130 | 134 | 129 | 133 | 131 | 135 |
| 136 | 140 | 138 | 142 | 137 | 141 | 139 | 143 |
| 144 | 148 | 146 | 150 | 145 | 149 | 147 | 151 |
| 152 | 156 | 154 | 158 | 153 | 157 | 155 | 159 |

Radio frame #1 Radio frame #2 ... Radio frame #8

FIG 3

Bit sequence

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ... 159

Row by row processing
8[4[2x2]x2]

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 |
| 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 |
| 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 144 | 145 | 146 | 147 | 48 | 149 | 150 | 151 |
| 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 4 | 2 | 6 | 1 | 5 | 3 | 7 |
| 8 | 12 | 10 | 14 | 9 | 13 | 11 | 15 |
| 16 | 20 | 18 | 22 | 17 | 21 | 19 | 23 |
| 24 | 28 | 26 | 30 | 25 | 29 | 27 | 31 |
| 32 | 36 | 34 | 38 | 33 | 37 | 35 | 39 |
| 40 | 44 | 42 | 46 | 41 | 45 | 43 | 47 |
| 48 | 52 | 50 | 54 | 49 | 53 | 51 | 55 |
| 56 | 60 | 58 | 62 | 57 | 61 | 59 | 63 |
| 64 | 68 | 66 | 70 | 65 | 69 | 67 | 71 |
| 72 | 76 | 74 | 78 | 73 | 77 | 75 | 79 |
| 80 | 84 | 82 | 86 | 81 | 85 | 83 | 87 |
| 88 | 92 | 90 | 94 | 89 | 93 | 91 | 95 |
| 96 | 100 | 98 | 102 | 97 | 101 | 99 | 103 |
| 104 | 108 | 106 | 110 | 105 | 109 | 107 | 111 |
| 112 | 116 | 114 | 118 | 113 | 117 | 115 | 119 |
| 120 | 124 | 122 | 126 | 121 | 125 | 123 | 127 |
| 128 | 132 | 130 | 134 | 129 | 133 | 131 | 135 |
| 136 | 140 | 138 | 142 | 137 | 141 | 139 | 143 |
| 144 | 148 | 146 | 150 | 145 | 149 | 147 | 151 |
| 152 | 156 | 154 | 158 | 153 | 157 | 155 | 159 |

Radio frame #1
Radio frame #2
...
Radio frame #8

FIG 4

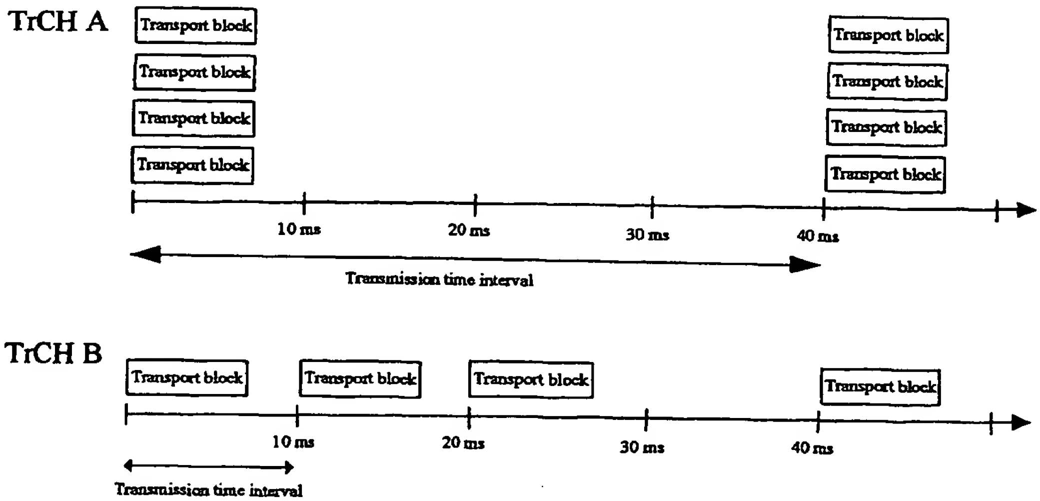
TrCH A
Transport block
Transport block
Transport block
Transport block
Transport block
Transport block
Transport block
Transport block
10 ms
20 ms
30 ms
40 ms
Transmission time interval
TrCH B
Transport block
Transport block
Transport block
Transport block
10 ms
20 ms
30 ms
40 ms
Transmission time interval

FIG 5

Input bit sequence: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 … 159

1st stage block Interleaving

Puncturing with simple shifting rule

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | **9** | 10 | 11 | 12 | 13 | **14** | 15 |
| 16 | 17 | **18** | 19 | 20 | 21 | 22 | **23** |
| 24 | 25 | 26 | **27** | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | **36** | 37 | 38 | 39 |
| **40** | 41 | 42 | 43 | 44 | **45** | 46 | 47 |
| 48 | **49** | 50 | 51 | 52 | 53 | **54** | 55 |
| 56 | 57 | **58** | 59 | 60 | 61 | 62 | **63** |
| 64 | 65 | 66 | **67** | 68 | 69 | 70 | 71 |
| 72 | 73 | 74 | 75 | **76** | 77 | 78 | 79 |
| **80** | 81 | 82 | 83 | 84 | **85** | 86 | 87 |
| 88 | **89** | 90 | 91 | 92 | 93 | **94** | 95 |
| 96 | 97 | **98** | 99 | 100 | 101 | 102 | **103** |
| 104 | 105 | 106 | **107** | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | **116** | 117 | 118 | 119 |
| **120** | 121 | 122 | 123 | 124 | **125** | 126 | 127 |
| 128 | **129** | 130 | 131 | 132 | 133 | **134** | 135 |
| 136 | 137 | **138** | 139 | 140 | 141 | 142 | **143** |
| 144 | 145 | 146 | **147** | 148 | 149 | 150 | 151 |
| 152 | 153 | 154 | 155 | **156** | 157 | 158 | 159 |

Column randomizing

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0** | 4 | 2 | 6 | 1 | 5 | 3 | 7 |
| 8 | 12 | 10 | **14** | **9** | 13 | 11 | 15 |
| 16 | 20 | **18** | 22 | 17 | 21 | 19 | **23** |
| 24 | 28 | 26 | 30 | 25 | 29 | **27** | 31 |
| 32 | **36** | 34 | 38 | 33 | 37 | 35 | 39 |
| **40** | 44 | 42 | 46 | 41 | **45** | 43 | 47 |
| 48 | 52 | 50 | **54** | **49** | 53 | 51 | 55 |
| 56 | 60 | **58** | 62 | 57 | 61 | 59 | **63** |
| 64 | 68 | 66 | 70 | 65 | 69 | **67** | 71 |
| 72 | **76** | 74 | 78 | 73 | 77 | 75 | 79 |
| **80** | 84 | 82 | 86 | 81 | **85** | 83 | 87 |
| 88 | 92 | 90 | **94** | **89** | 93 | 91 | 95 |
| 96 | 100 | **98** | 102 | 97 | 101 | 99 | **103** |
| 104 | 108 | 106 | 110 | 105 | 109 | **107** | 111 |
| 112 | **116** | 114 | 118 | 113 | 117 | 115 | 119 |
| **120** | 124 | 122 | 126 | 121 | **125** | 123 | 127 |
| 128 | 132 | 130 | **134** | **129** | 133 | 131 | 135 |
| 136 | 140 | **138** | 142 | 137 | 141 | 139 | **143** |
| 144 | 148 | 146 | 150 | 145 | 149 | **147** | 151 |
| 152 | **156** | 154 | 158 | 153 | 157 | 155 | 159 |

Frame #1 Frame #2 … Frame #8

FIG 6

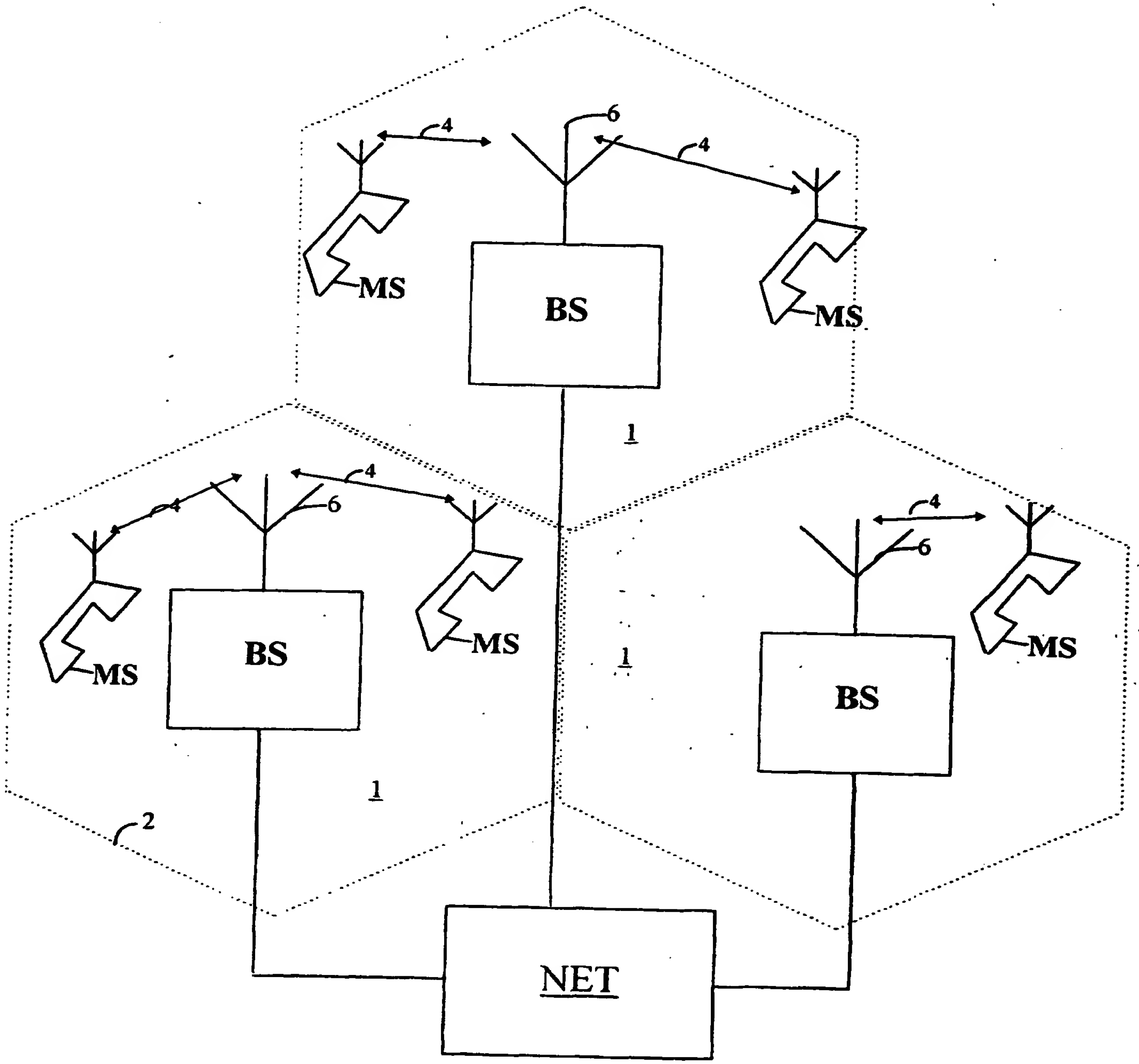


FIG 7

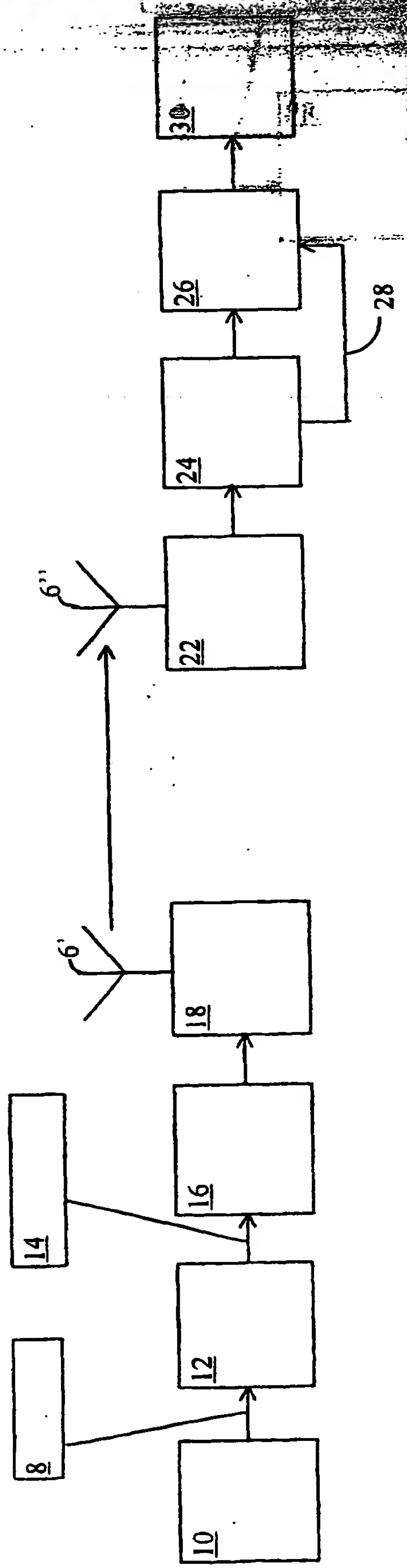


FIG 8

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | **4** | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | **13** | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | **22** | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | **31** |
| **32** | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | **41** | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | **50** | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | **59** | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | **68** | 69 | 70 | 71 |
| 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | **86** | 87 |
| 88 | 89 | 90 | 91 | 92 | 93 | 94 | **95** |
| **96** | 97 | 98 | 99 | 100 | 101 | 102 | 103 |
| 104 | **105** | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | **114** | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | **123** | 124 | 125 | 126 | 127 |

FIG 9

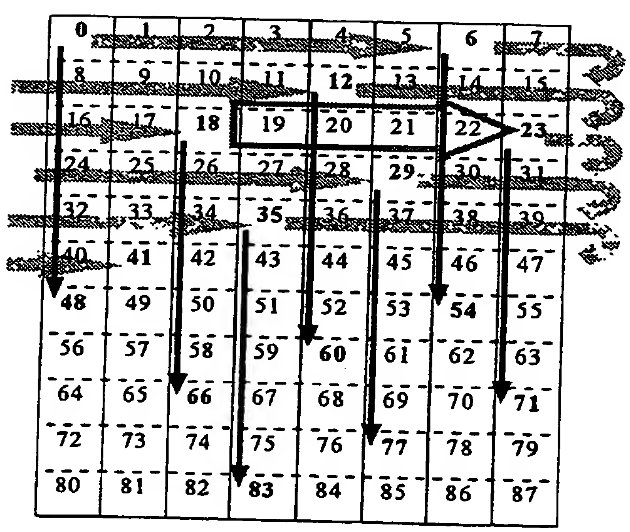


FIG 10

| S ; T | K | 1 | 2 | | 4 | | | | 8 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | k | 0 | 0 | 1 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Q | 1 | 0;0 | 0;0 | 0;1 | 0;0 | 0;2 | 0;1 | 0;3 | 0;0 | 0;4 | 0;2 | 0;6 | 0;1 | 0;5 | 0;3 | 0;7 |
| | 2 | | 0;0 | 1;1 | 0;0 | 0;1 | 1;3 | 0;2 | 0;0 | 0;2 | 0;1 | 0;3 | 1;5 | 1;7 | 1;6 | 0;4 |
| | 3 | | | | 0;0 | 1;2 | 2;3 | 0;1 | 0;0 | 1;4 | 2;6 | 0;2 | 1;3 | 2;7 | 0;1 | 1;5 |
| | 4 | | | | 0;0 | 1;2 | 2;3 | 0;1 | 0;0 | 0;1 | 2;5 | 1;4 | 3;7 | 2;6 | 1;3 | 0;2 |
| | 5 | | | | | | | | 0;0 | 3;4 | 2;2 | 4;6 | 4;5 | 1;1 | 5;7 | 2;3 |
| | 6 | | | | | | | | 0;0 | 1;2 | 2;3 | 0;1 | 5;7 | 3;5 | 4;6 | 2;4 |
| | 7 | | | | | | | | 0;0 | 3;4 | 5;6 | 1;2 | 6;7 | 2;3 | 4;5 | 0;1 |
| | 8 | | | | | | | | 0;0 | 3;4 | 5;6 | 1;2 | 6;7 | 2;3 | 4;5 | 0;1 |

FIG 11

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | **12** | 13 | 14 | 15 |
| 16 | **17** | 18 | 19 | 20 | 21 | **22** | 23 |
| 24 | 25 | 26 | **27** | 28 | 29 | 30 | 31 |
| **32** | 33 | 34 | 35 | 36 | **37** | 38 | 39 |
| 40 | 41 | **42** | 43 | 44 | 45 | 46 | **47** |
| 48 | 49 | 50 | 51 | **52** | 53 | 54 | 55 |
| 56 | **57** | 58 | 59 | 60 | 61 | **62** | 63 |
| 64 | 65 | 66 | **67** | 68 | 69 | 70 | 71 |
| **72** | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | **82** | 83 | 84 | 85 | 86 | **87** |
| 88 | 89 | 90 | 91 | **92** | 93 | 94 | 95 |
| 96 | **97** | 98 | 99 | 100 | 101 | **102** | 103 |
| 104 | 105 | 106 | **107** | 108 | 109 | 110 | 111 |
| **112** | 113 | 114 | 115 | 116 | **117** | 118 | 119 |
| 120 | 121 | **122** | 123 | 124 | 125 | 126 | **127** |
| 128 | 129 | 130 | 131 | **132** | 133 | 134 | 135 |
| 136 | **137** | 138 | 139 | 140 | 141 | **142** | 143 |
| 144 | 145 | 146 | **147** | 48 | 149 | 150 | 151 |
| **152** | 153 | 154 | 155 | 156 | **157** | 158 | 159 |

FIG 12

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | **3** | 4 | 5 | 6 | 7 |
| 8 | 9 | **10** | 11 | 12 | 13 | 14 | 15 |
| 16 | **17** | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| **32** | 33 | 34 | 35 | 36 | 37 | 38 | **39** |
| 40 | 41 | 42 | 43 | 44 | 45 | **46** | 47 |
| 48 | 49 | 50 | 51 | 52 | **53** | 54 | 55 |
| 56 | 57 | 58 | 59 | **60** | 61 | 62 | 63 |
| 64 | 65 | 66 | **67** | 68 | 69 | 70 | 71 |
| 72 | 73 | **74** | 75 | 76 | 77 | 78 | 79 |
| 80 | **81** | 82 | 83 | 84 | 85 | 86 | 87 |
| 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| **96** | 97 | 98 | 99 | 100 | 101 | 102 | **103** |
| 104 | 105 | 106 | 107 | 108 | 109 | **110** | 111 |
| 112 | 113 | 114 | 115 | 116 | **117** | 118 | 119 |
| 120 | 121 | 122 | 123 | **124** | 125 | 126 | 127 |

FIG 13

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 |
| 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 |
| 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | | |

FIG 14

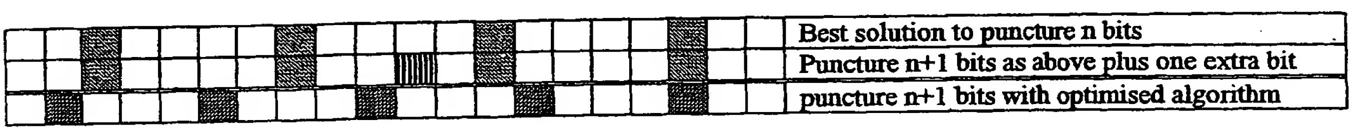


FIG 15

EPO - Munich
17
19. März 1999



Abstract

Data communications apparatus and method of communicating data

Near optimum puncturing patterns are possible when applying rate matching after first interleaving. The necessary algorithm is not very complex, it is similar to the puncturing algorithm itself but has to be executed once per frame only, not once per bit.

Fig 15

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record**

# BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)